NBdata Models

Marina Peñalver Ripoll

Contents

Introduction

In this document, we will use the `maSigPro` package to analyze the evolution of one gene.

First, we load the necessary libraries for the analysis.

We load the dataset provided by the `maSigPro` package, which includes the expression data.

```
data(NBdata)
data(NBdesign)
```

Create the design matrix from the experimental design.

```
d <- make.design.matrix(NBdesign)
design <- as.data.frame(NBdesign)
```

We define functions to plot the expression of specific genes across time points for two groups.

```
plot_gene <- function(gene_number, design, NBdata) {
  gene_data <- data.frame(
    Time = design$Time[1:18],
    Group1 = NBdata[gene_number, 1:18],
    Group2 = NBdata[gene_number, 19:36]
  )

  data_mean <- data.frame(
    Time = c(0, 12, 24, 36, 48, 60),
    Mean.G1 = sapply(split(gene_data$Group1, gene_data$Time), mean),
    Mean.G2 = sapply(split(gene_data$Group2, gene_data$Time), mean)
  )

  plot <- ggplot(gene_data) +
    geom_point(data = gene_data, aes(x = Time, y = Group1, color = "Group 1")) +
    geom_point(data = gene_data, aes(x = Time, y = Group2, color = "Group 2")) +
    geom_line(data = data_mean, aes(x = Time, y = Mean.G1), linewidth = 1, color = "coral") +
    geom_line(data = data_mean, aes(x = Time, y = Mean.G2), linewidth = 1, color = "steelblue2") +
    labs(title = paste("Gene", gene_number),
         x = "Time",
         y = "Expression",
         color = NULL) +
    scale_color_manual(values = c("Group 1" = "coral", "Group 2" = "steelblue2")) +
    theme_light()
}

plot_genes <- function(gene_numbers, design, NBdata, ncol, nrow) {
  plots <- list()
  j = 1
  for (i in gene_numbers) {
    plots[[j]] <- plot_gene(i, design, NBdata)
    j = j + 1
  }
  combined_plot <- wrap_plots(plots, ncol = ncol, nrow = nrow)
  return(combined_plot)
}
```

And we define some functions for plotting the models:

```
plot_lm <- function(model, design=design, title=NULL){
  data <- data.frame(
    Group1 = model$fitted.values[1:18],
    Group2 = model$fitted.values[19:36],
    Time = design$Time[1:18]
  )

  n = 200
  time2 <- seq(from = 0, to = 60, length.out = n)
  data2 <- data.frame(
    time2 = time2,
    pred.G1 = predict(model, data.frame(Time = time2, Group=as.factor(rep("Group.1", n)))),
    pred.G2 = predict(model, data.frame(Time = time2, Group=as.factor(rep("Group.2", n))))
  )

  plot <- ggplot(data) +
    geom_point(data = gene_data, aes(x = Time, y = Group1, color = "Grupo 1"), size=2.5, shape=18) +
    geom_point(data = gene_data, aes(x = Time, y = Group2, color = "Grupo 2"), size=2.5, shape=18) +
    geom_point(aes(x = Time, y = Group1, color = "Grupo 1"), size=3) +
    geom_point(aes(x = Time, y = Group2, color = "Grupo 2"), size=3) +
    geom_line(data=data2, aes(x = time2, y = pred.G1), linewidth = 1, color="coral") +
    geom_line(data=data2, aes(x = time2, y = pred.G2), linewidth = 1, color="steelblue2") +
    labs(title = title,
         x = "Tiempo",
         y = "Expresión",
         color = NULL) +
    scale_color_manual(values = c("Grupo 1" = "coral", "Grupo 2" = "steelblue2")) +
    theme_light()

  return(plot)
}
```

```
plot_glm <- function(model, design=design, title=NULL){
  data <- data.frame(
    Group1 = model$fitted.values[1:18],
    Group2 = model$fitted.values[19:36],
    Time = design$Time[1:18]
  )

  n = 200
  time2 <- seq(from = 0, to = 60, length.out = n)
  data2 <- data.frame(
    time2 = time2,
    pred.G1 = predict(model, data.frame(Time = time2, Group=as.factor(rep("Group.1", n))), type="respons
    pred.G2 = predict(model, data.frame(Time = time2, Group=as.factor(rep("Group.2", n))), type="respons
  )

  plot <- ggplot(data) +
    geom_point(data = gene_data, aes(x = Time, y = Group1, color = "Grupo 1"), size=2.5, shape=18) +
    geom_point(data = gene_data, aes(x = Time, y = Group2, color = "Grupo 2"), size=2.5, shape=18) +
    geom_point(aes(x = Time, y = Group1, color = "Grupo 1"), size=3) +
    geom_point(aes(x = Time, y = Group2, color = "Grupo 2"), size=3) +
    geom_line(data=data2, aes(x = time2, y = pred.G1), linewidth = 1, color="coral") +
    geom_line(data=data2, aes(x = time2, y = pred.G2), linewidth = 1, color="steelblue2") +
    labs(title = title,
         x = "Tiempo",
         y = "Expresión",
         color = NULL) +
    scale_color_manual(values = c("Grupo 1" = "coral", "Grupo 2" = "steelblue2")) +
    theme_light()

  return(plot)
}
```

Linear and Polynomial Regression

We fit linear and polynomial regression models to the data and plot the results.

```
# Select a gene for analysis
gen <- 13
y <- as.numeric(NBdata[gen,])
Time <- NBdesign[,1]
Group <- factor(NBdesign[, 3] + 2 * NBdesign[, 4],
                labels = colnames(NBdesign[, 3:4]))

data <- data.frame(
  y = y,
  Time = Time,
  Group = Group
)

gene_data <- data.frame(
  Time = design$Time[1:18],
  Group1 = NBdata[gen, 1:18],
  Group2 = NBdata[gen, 19:36]
)
```

```
# Linear regression models
lm1 <- lm(y ~ Time + Group)
lm2 <- lm(y ~ Time * Group)
# summary(lm1)
# summary(lm2)
anova(lm1, lm2)
```

```
## Analysis of Variance Table
##
## Model 1: y ~ Time + Group
## Model 2: y ~ Time * Group
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1     33 77777
## 2     32 54674  1     23103 13.522 0.0008591 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Polynomial regression models
lm3 <- lm(y ~ poly(Time, 2) + Group)
lm4 <- lm(y ~ poly(Time, 2) * Group)
# summary(lm3)
# summary(lm4)
anova(lm3, lm4)
```

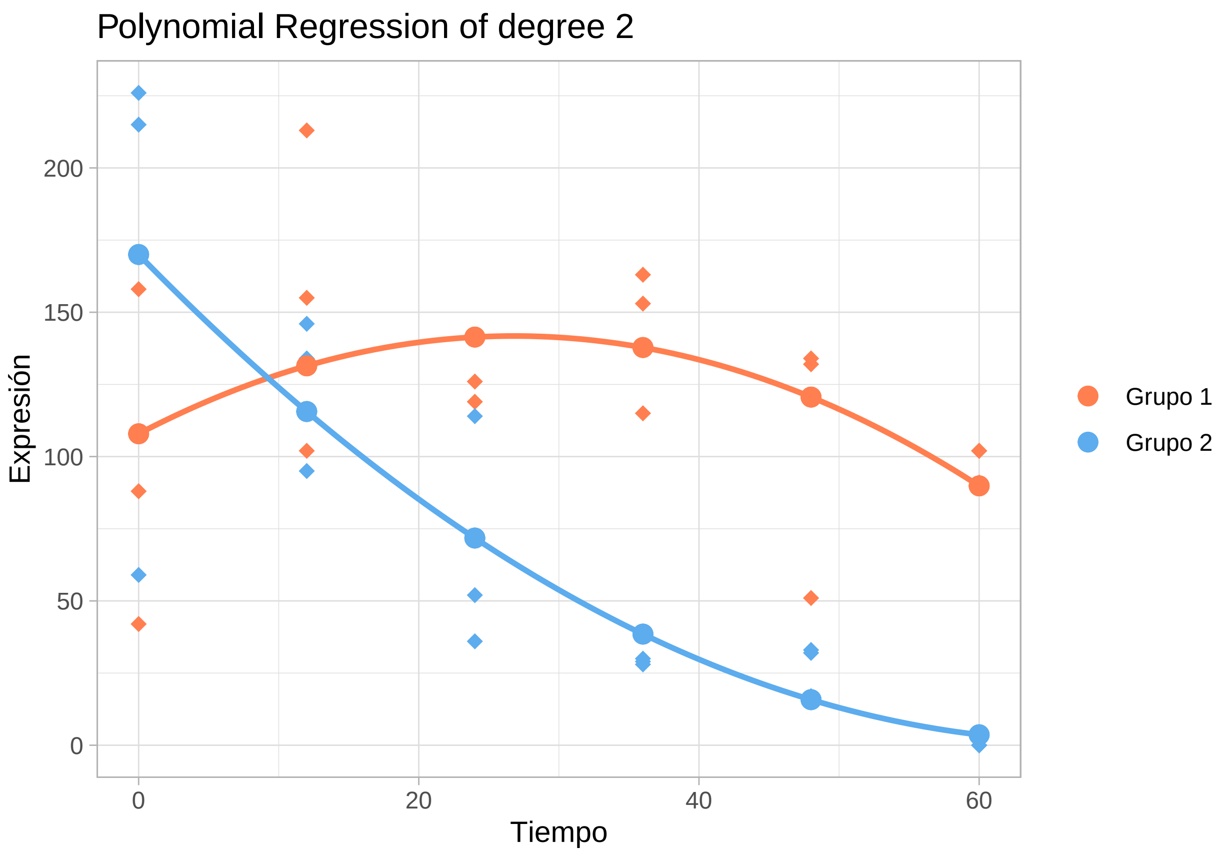
```
## Analysis of Variance Table
##
## Model 1: y ~ poly(Time, 2) + Group
## Model 2: y ~ poly(Time, 2) * Group
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1     32 77650
## 2     30 46395  2     31255 10.105 0.0004415 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can search the optimal degree for polynomial regression:

```
max_degree <- 5
aic_values <- numeric(max_degree)
bic_values <- numeric(max_degree)
sce_values <- numeric(max_degree)

for (degree in 1:max_degree) {
  model <- lm(y ~ poly(Time, degree) * Group)
  aic_values[degree] <- AIC(model)
  bic_values[degree] <- BIC(model)
  sce_values[degree] <- sum((model$fitted.values - y)^2)
}

optimal_degree <- which.min(aic_values)
optimal_lm <- lm(y ~ poly(Time, optimal_degree) * Group)
print(plot_lm(optimal_lm, design = design,
              title = paste("Polynomial Regression of degree", optimal_degree)))
```



Generalized Linear Models (GLM)

We fit GLM models to the data and plot the results.

```
glm1 <- glm(y ~ Time + Group, family = poisson)
glm2 <- glm(y ~ Time * Group, family = poisson)
# summary(glm1)
# summary(glm2)
anova(glm1, glm2, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: y ~ Time + Group
## Model 2: y ~ Time * Group
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1        33    1070.43
## 2        32     532.69  1   537.73 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

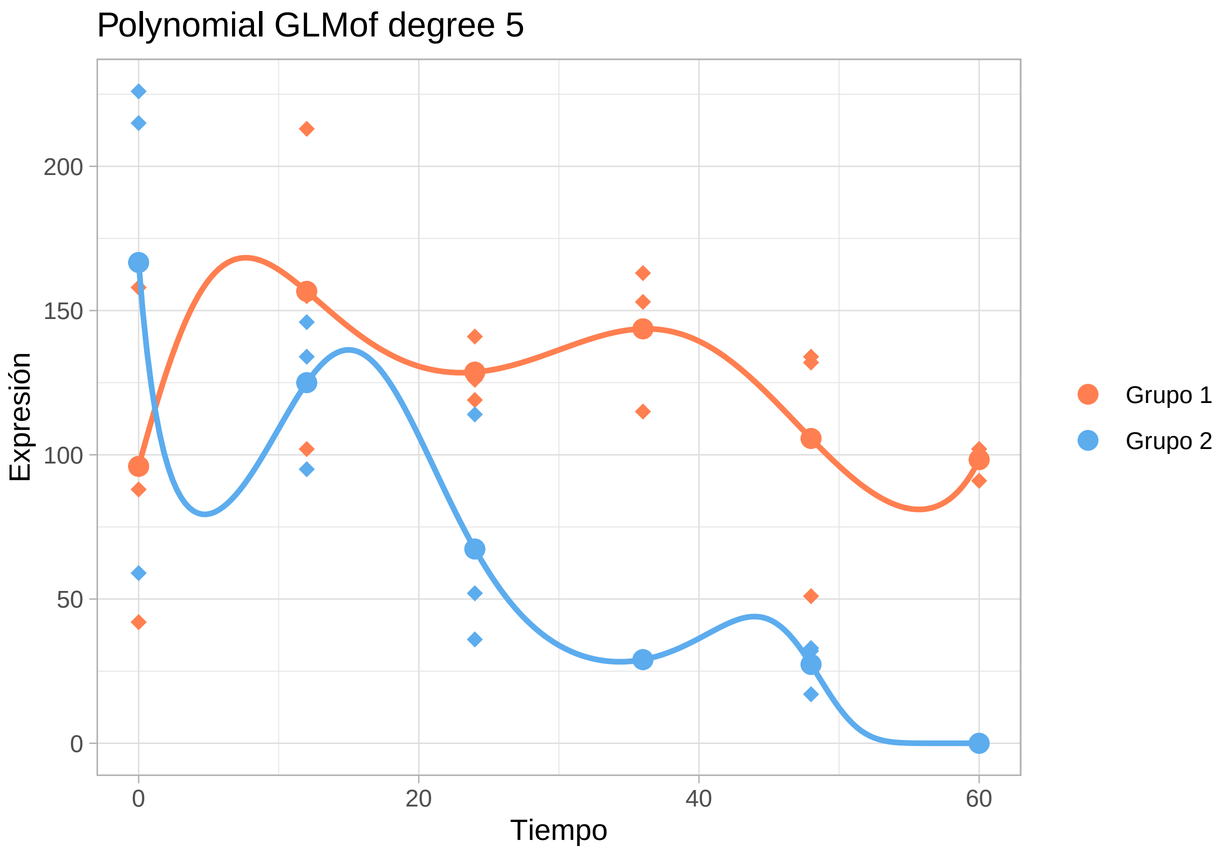
```
# Polynomial GLM models
glm3 <- glm(y ~ poly(Time, 2) + Group, family = poisson)
glm4 <- glm(y ~ poly(Time, 2) * Group, family = poisson)
# summary(glm3)
# summary(glm4)
anova(glm3, glm4, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: y ~ poly(Time, 2) + Group
## Model 2: y ~ poly(Time, 2) * Group
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1        32    1053.82
## 2        30     454.21  2   599.61 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Optimal degree for polynomial GLM
max_degree <- 5
aic_values <- numeric(max_degree)
bic_values <- numeric(max_degree)
sce_values <- numeric(max_degree)

for (degree in 1:max_degree) {
  model <- glm(y ~ poly(Time, degree) * Group, family = poisson)
  aic_values[degree] <- AIC(model)
  bic_values[degree] <- BIC(model)
  sce_values[degree] <- sum((model$fitted.values - y)^2)
}

optimal_degree <- which.min(aic_values)
optimal_glm <- glm(y ~ poly(Time, optimal_degree) * Group, family = poisson)
print(plot_glm(optimal_glm, design = design,
               title = paste("Polynomial GLMof degree", optimal_degree)))
```



Generalized Additive Models (GAM)

We fit GAM models to the data using natural and cubic splines, and plot the results.

```r
# Function to evaluate GAM models
evaluate_gam <- function(y, Time, Group, knot_range, family = poisson(), type = "natural") {
  aic_values <- numeric(length(knot_range))
  bic_values <- numeric(length(knot_range))
  models <- vector("list", length(knot_range))
  sce <- numeric(length(knot_range))

  for (i in seq_along(knot_range)) {
    k <- knot_range[i]
    if (type == "natural") {
      gam_model <- gam(y ~ ns(Time, df = k + 1) * Group, family = family,
                       data = data.frame(Time = Time, Group = Group, y = y)) }
    else if (type == "cubic") {
      gam_model <- gam(y ~ bs(Time, df = k + 3) * Group, family = family,
                       data = data.frame(Time = Time, Group = Group, y = y)) }
    aic_values[i] <- AIC(gam_model)
    bic_values[i] <- BIC(gam_model)
    models[[i]] <- gam_model
    sce[i] <- sum((gam_model$fitted.values - y)^2)
  }

  optimal_index <- which.min(aic_values)
  optimal_knots <- knot_range[optimal_index]
  optimal_model <- models[[optimal_index]]

  return(list(optimal_knots = optimal_knots, optimal_model = optimal_model, aic_values = aic_values, sc
}

knot_range = 1:4
```

Cubic spline

We evaluate GAM models with cubic spline:

```r
evaluate_gam(y, Time, Group, knot_range, type = "cubic")
```

```
## $optimal_knots
## [1] 2
##
## $optimal_model
##
## Family: poisson
## Link function: log
##
## Formula:
## y ~ bs(Time, df = k + 3) * Group
## Total model degrees of freedom 12
##
## UBRE score: 9.771705
##
## $aic_values
## [1] 630.5454 595.2002 595.2002 595.2002
##
## $sce
## [1] 44353.06 41517.33 41517.33 41517.33
```

And we fit the model with 2 knots

```
optimal_cubic_gam <- gam(y ~ bs(Time, df = 2 + 3) * Group, family = poisson, data = data)
```

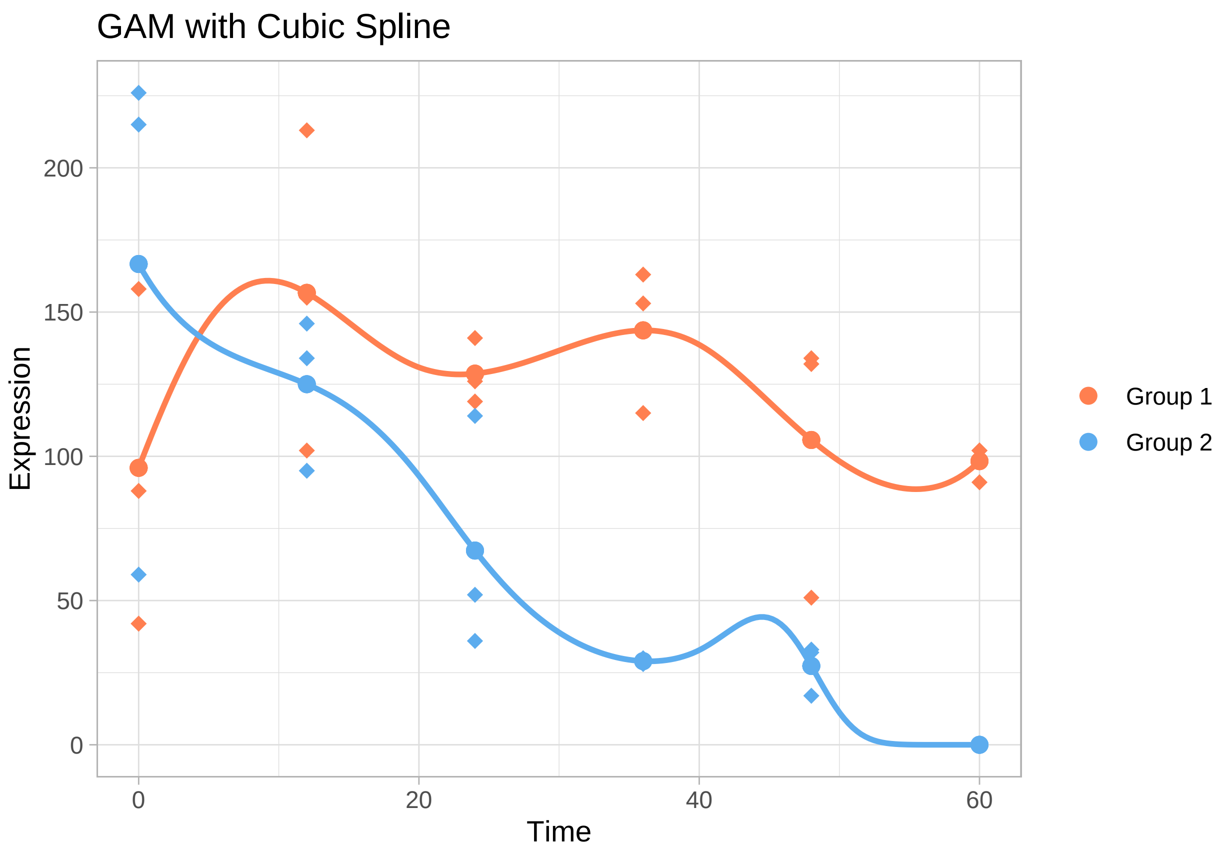
We can visualize the model:

```
n <- 200
time2 <- seq(from = 0, to = 60, length.out = n)
data_bs1 <- data.frame(Time = time2, Group = as.factor(rep("Group.1", n)))
data_bs1 <- cbind(data_bs1, bs(time2, df = 2 + 3))
data_bs2 <- data.frame(Time = time2, Group = as.factor(rep("Group.2", n)))
data_bs2 <- cbind(data_bs2, bs(time2, df = 2 + 3))

data1 <- data.frame(
  Group1 = optimal_cubic_gam$fitted.values[1:18],
  Group2 = optimal_cubic_gam$fitted.values[19:36],
  Time = design$Time[1:18])

data2 <- data.frame(
  time2 = time2,
  pred.G1 = predict(optimal_cubic_gam, newdata = data_bs1, type = "response"),
  pred.G2 = predict(optimal_cubic_gam, newdata = data_bs2, type = "response"))

ggplot(data1) +
  geom_point(data = gene_data, aes(x = Time, y = Group1, color = "Group 1"), size = 2.5, shape = 18) +
  geom_point(data = gene_data, aes(x = Time, y = Group2, color = "Group 2"), size = 2.5, shape = 18) +
  geom_point(aes(x = Time, y = Group1, color = "Group 1"), size = 2.5) +
  geom_point(aes(x = Time, y = Group2, color = "Group 2"), size = 2.5) +
  geom_line(data = data2, aes(x = time2, y = pred.G1), linewidth = 1, color = "coral") +
  geom_line(data = data2, aes(x = time2, y = pred.G2), linewidth = 1, color = "steelblue2") +
  labs(title = "GAM with Cubic Spline",
       x = "Time",
       y = "Expression",
       color = NULL) +
  scale_color_manual(values = c("Group 1" = "coral", "Group 2" = "steelblue2")) +
  theme_light()
```



Natural cubic splines

Now, we will use natural cubic spline in our model

```
evaluate_gam(y, Time, Group, knot_range, type = "natural")
```

```
## $optimal_knots
## [1] 4
##
## $optimal_model
##
## Family: poisson
## Link function: log
##
## Formula:
## y ~ ns(Time, df = k + 1) * Group
## Total model degrees of freedom 12
##
## UBRE score: 9.771705
##
## $aic_values
## [1] 679.5799 674.7023 635.9420 595.2002
##
## $sce
## [1] 46495.47 45952.75 44605.15 41517.33
```

The optimal model has 4 knots:

```r
optimal_natural_gam <- gam(y ~ ns(Time, df = 4+1) * Group, family = poisson, data = data)
```

The graph is:

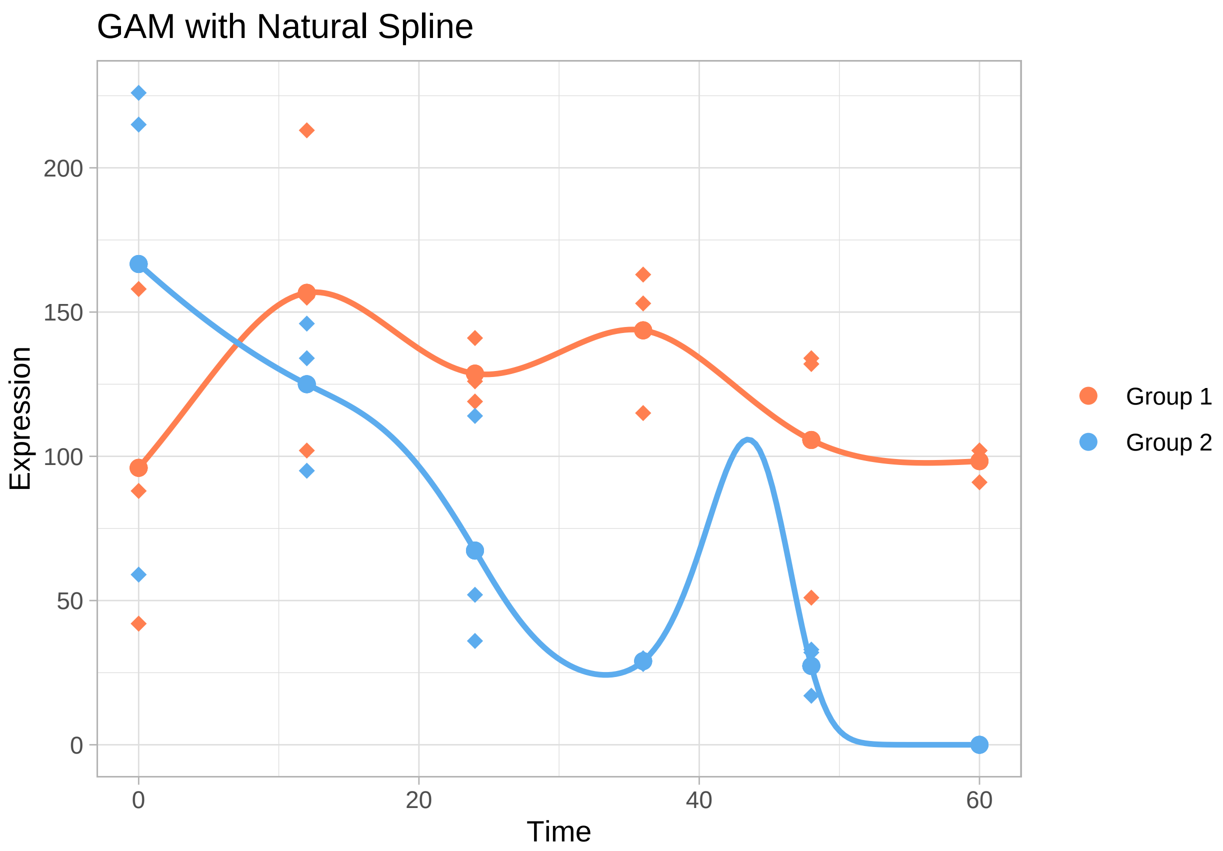
```r
n <- 200
time2 <- seq(from = 0, to = 60, length.out = n)
data_ns1 <- data.frame(Time = time2, Group = as.factor(rep("Group.1", n)))
data_ns1 <- cbind(data_ns1, ns(time2, df = 4 + 1))
data_ns2 <- data.frame(Time = time2, Group = as.factor(rep("Group.2", n)))
data_ns2 <- cbind(data_ns2, ns(time2, df = 4 + 1))

data1 <- data.frame(
  Group1 = optimal_natural_gam$fitted.values[1:18],
  Group2 = optimal_natural_gam$fitted.values[19:36],
  Time = design$Time[1:18])

data2 <- data.frame(
  time2 = time2,
  pred.G1 = predict(optimal_natural_gam, newdata = data_ns1, type

 = "response"),
  pred.G2 = predict(optimal_natural_gam, newdata = data_ns2, type = "response"))

ggplot(data1) +
  geom_point(data = gene_data, aes(x = Time, y = Group1, color = "Group 1"), size = 2.5, shape = 18) +
  geom_point(data = gene_data, aes(x = Time, y = Group2, color = "Group 2"), size = 2.5, shape = 18) +
  geom_point(aes(x = Time, y = Group1, color = "Group 1"), size = 2.5) +
  geom_point(aes(x = Time, y = Group2, color = "Group 2"), size = 2.5) +
  geom_line(data = data2, aes(x = time2, y = pred.G1), linewidth = 1, color = "coral") +
  geom_line(data = data2, aes(x = time2, y = pred.G2), linewidth = 1, color = "steelblue2") +
  labs(title = "GAM with Natural Spline",
       x = "Time",
       y = "Expression",
       color = NULL) +
  scale_color_manual(values = c("Group 1" = "coral", "Group 2" = "steelblue2")) +
  theme_light()
```


GAM with Natural Spline
Expression
200
150
100
50
0
0
20
40
60
Time
Group 1
Group 2